Research Paper ■

Methods for Semi-automated Indexing for High Precision Information Retrieval

Daniel C. Berrios, MD, PhD, MPH, Russell J. Cucina, MD, Lawrence M. Fagan, MD, PhD

Abstract **Objective**. To evaluate a new system, ISAID (Internet-based Semi-automated Indexing of Documents), and to generate textbook indexes that are more detailed and more useful to readers.

Design. Pilot evaluation: simple, nonrandomized trial comparing ISAID with manual indexing methods. Methods evaluation: randomized, cross-over trial comparing three versions of ISAID and usability survey.

Participants. Pilot evaluation: two physicians. Methods evaluation: twelve physicians, each of whom used three different versions of the system for a total of 36 indexing sessions.

Measurements. Total index term tuples generated per document per minute (TPM), with and without adjustment for concordance with other subjects; inter-indexer consistency; ratings of the usability of the ISAID indexing system.

Results. Compared with manual methods, ISAID decreased indexing times greatly. Using three versions of ISAID, inter-indexer consistency ranged from 15% to 65% with a mean of 41%, 31%, and 40% for each of three documents. Subjects using the full version of ISAID were faster (average TPM: 5.6) and had higher rates of concordant index generation. There were substantial learning effects, despite our use of a training/run-in phase. Subjects using the full version of ISAID were much faster by the third indexing session (average TPM: 9.1). There was a statistically significant increase in three-subject concordant indexing rate using the full version of ISAID during the second indexing session ($p < 0.05$).

Summary. Users of the ISAID indexing system create complex, precise, and accurate indexing for full-text documents much faster than users of manual methods. Furthermore, the natural language processing methods that ISAID uses to suggest indexes contributes substantially to increased indexing speed and accuracy.

■ **J Am Med Inform Assoc.** 2002;9:637–652. DOI 10.1197/jamia.M1075.

Affiliations of the authors: Stanford Medical Informatics, Stanford University, Stanford, California.

This study was supported in part by the Veterans Affairs Office of Academic Affairs and Health Services Research, Development Service Research Funds, Office of the Chief Information Officer; the Department of the Army, Cooperative Agreement Number (DAMD17-97-2-7016); the Center for Total Access, Fort Gordon, Georgia; and the Telemedicine and Advanced Technology Research Center, Fort Detrick, Maryland.

Correspondence and reprints: Daniel C. Berrios, MD, PhD, MPH, Research Institute for Advanced Computer Science, Ames Research Center, NASA, MailStop 269-2, Moffett Field, CA 94035; e-mail: <berrios@email.arc.nasa.gov>.

Received for publication: 1/11/02; accepted for publication: 6/14/02.

The number of textbooks related to patient care and basic science is already large and continues to grow. In areas of health care with rapidly evolving or intricate management strategies, textbooks constitute a critical resource for health providers. In 1991, Forsythe and colleagues observed clinicians on rounds and in clinics, recording 454 clinical questions that arose and the ultimate sources of answers to these queries.[1] The clinicians obtained answers to many questions by consulting multiple information sources, most often the patient's medical chart or records or the patient himself. Answering a quarter of the questions, however, required consulting medical library materials. Of

these, the answers to more than half were found in a textbook and the remainder in medical journals. In another study, internists' self-reported use of various information sources roughly matched these figures.[2] During outpatient treatment of AIDS patients, five textbooks, in conjunction with a system for searching medical journals, provided answers to approximately 75% of clinicians' questions.[3]

However, the use of textbooks is frequently neither straightforward nor expedient. Searching a textbook consumes on average six minutes of time that could be used for other clinical care tasks.[4] Faced with an urgent information need, clinicians often must rely on manual inspection of a table of contents or alphabetized keyword index to guide their search. Although formal evaluations of such indexes are lacking, in one study only 30% of internists felt that textbooks are "adequately indexed for rapid information retrieval."[2] Tables of content most commonly list only one or two levels of section headings, each indexed by the page number on which the section begins. Alphabetized keyword indexes, like those commonly found at the end of most textbooks, may cross-reference two or more words that occur in the textbook and therefore are usually more precise. However, like tables of content, most of these indexes also direct the reader only to whole pages in the textbook; readers are still left with the time-consuming task of finding specific answers to their questions in the text of those pages.

Many textbooks have recently become available in electronic format (e.g., on CD-ROM). Some of these textbooks can now be searched by locating terms that occur in the text, either alone or in some predefined proximity to each other. However, this type of indexing alone is unlikely to improved textbook search precision. Hersh found the precision of information retrieval from one medical textbook using such term search methods to be an abysmal 19%.[5] In addition, the organization of some electronic medical textbooks is such that low cognitive load tasks (e.g., visual scanning) cannot be performed as easily as with printed versions, even with key term highlighting.[6]

Indexes that would allow clinicians, researchers, and patients to retrieve the information they need from these sources rapidly and with greater precision must contain more knowledge than merely the location of the beginning of textbook sections or the numbers of pages on which one or two concepts are discussed. Entries in these indexes must mirror the questions that drive readers to use the textbook to seek knowledge. Furthermore, these indexes must point the reader to more specific locations in the text. For example, consider a resident physician with the specific question "What is the appropriate duration of therapy for the treatment of a patient with *Pseudomonas* pneumonia using aminoglycosides?" A traditional alphabetized keyword index might contain an entry for "pneumonia" and "therapy" that points to several different pages in a textbook, only a minority of which contain discussions of the length of treatment of pneumonia caused by *Pseudomonas* species. A superior index would allow residents first to find their exact question in the index and then to find the specific portions of a textbook that contain answers.

Both the creation and the use of such detailed and specific indexes present several challenges. First, they are potentially much larger than traditional keyword indexes, because they must include entries for large numbers of different and specific, complex questions readers have, and the specific locations of answers to these questions. Indeed, for many textbooks, such indexes may be so large that they themselves require a system for navigating to a desired question in the index to be of any practical use. Second, the amount of labor required to generate such indexes manually is likely to be very large, because more extensive coordination of indexed terms and more specific reference back into the text are required. Finally, the nature of such detailed, query-based indexes requires that indexers have significant domain-specific knowledge, particularly an understanding of the proper and specific relation between index terms and the ability to recognize index terms in the text that are implied, but not stated.

We have developed ELBook, a computer-based system for retrieving fine-grained (i.e., highly specific) information from text documents.[7] ELBook requires that domain experts make explicit as indexes some of the knowledge contained in the text of documents. However, unlike the pioneering Hepatitis Knowledge-Base,[8] ELBook does not require indexers to provide a representation of all the knowledge contained in its documents. Query and concept models constrain the space of possible ELBook indexes and queries. Thus, the design of ELBook builds on the attempts by Hersh, Purcell,[9] and others to incorporate more domain-specific knowledge into full-text IR systems (Table 1).

Enthusiasm for the retrieval capabilities of ELBook has consistently been tempered, however, by the realization that generating indexes for ELBook without

Table 1 ■

Full-text Information Retrieval Systems and Selected Characteristics of Their Indexing Methods

			Index		
System	Unit of Retrieval	Query Formulation	Schema	Scope	Generation
SMART[31]	Full document	Natural Language	Natural Language	Entire Document	Automated
FOLIOviews	Paragraph	Natural Language	Natural Language	Paragraph	Automated
Context-Based Search[9]	Full Document	Context Model & Natural Language	Context Model with Natural Language	Document Context	Manual & Automated
SAPHIRE[5]	Textbook Section	UMLS concepts	UMLS concepts	Textbook Section	Automated
ELBook[7]	Text Element (e.g., a sentence)	Query and Concept Models	UMLS concepts and Natural Language	Text Element (e.g., a sentence)	Semi-Automated

computer-based support is extremely time-consuming.[10] Furthermore, to evaluate the performance of ELBook on any test collection, we require a substantial amount of such precise indexing. If indexes of this type cannot be generated with sufficient accuracy and in an acceptably efficient manner, increases in search precision or recall using these indexes would be moot. Thus, we set out to develop a computer-based system that could provide automated support for indexing full-text documents for use in ELBook.

Several researchers have attempted to automate some or all of the process of generating indexes for various types of full-text documents. Investigators at the National Library of Medicine (NLM) have described both semi-automated[11] and fully automated[12] indexing systems designed for journal publications. MedLEE, a natural language understanding system, can extract concepts from clinical notes and reports with reasonable accuracy that can then be used as indexes, although modeling domain knowledge for specific applications remains a bottleneck.[13–16] Furthermore, NLP systems like MedLEE typically do not provide support for interactive document indexing; such interaction could improve the accuracy of index generation through human review.

This article evaluates ISAID (Internet-based Semi-automated Indexing of Documents), a computer-based system to generate textbook indexes that are more detailed and, hopefully, more useful to readers. This system requires domain-dependent query and concept models as well as a domain-independent document model to provide some of the knowledge required to create such complex indexes. ISAID requires that a domain expert first describe a set of questions, or generic queries, to be used as templates for indexes. Collectively, these questions constitute the query model. The concept model that ISAID uses for the medical domain is largely based on the Unified Medical Language System semantic network. The document model is based on the explicit and implicit structure of Hypertext Markup Language (HTML) documents. ISAID uses a modified vector-space model to help propose candidate indexes. We performed limited comparisons of the ability of ISAID users to generate indexes versus a manual indexing system, and then proceeded to evaluate the contributions of the document and vector-space models to the indexing process. We examined the consistency and speed of indexers using ISAID as a necessary first step towards the evaluation of ELBook.

Methods

ELBook (Figure 1) includes QueryEditor, a system for generating query models required for both the indexing and retrieval of HTML document elements, and a search system that matches indexes and user queries. The query models completely constrain the space of document indexes generated with ISAID, and allow indexers to precisely specify relations between index terms. After these indexes are generated, the same query models constrain possible ELBook queries to this same space. Thus, the precision of ELBook searches should be enhanced compared with traditional keyword searches because the relations between terms are specified and because the space of search queries is limited to the space of index terms and relations.

ISAID System Architecture

The ISAID indexing system consists of two separate components designed to work sequentially: a **text analyzer** (TA) that extracts and stores knowledge from documents and an **indexing interface** (Figure 2)

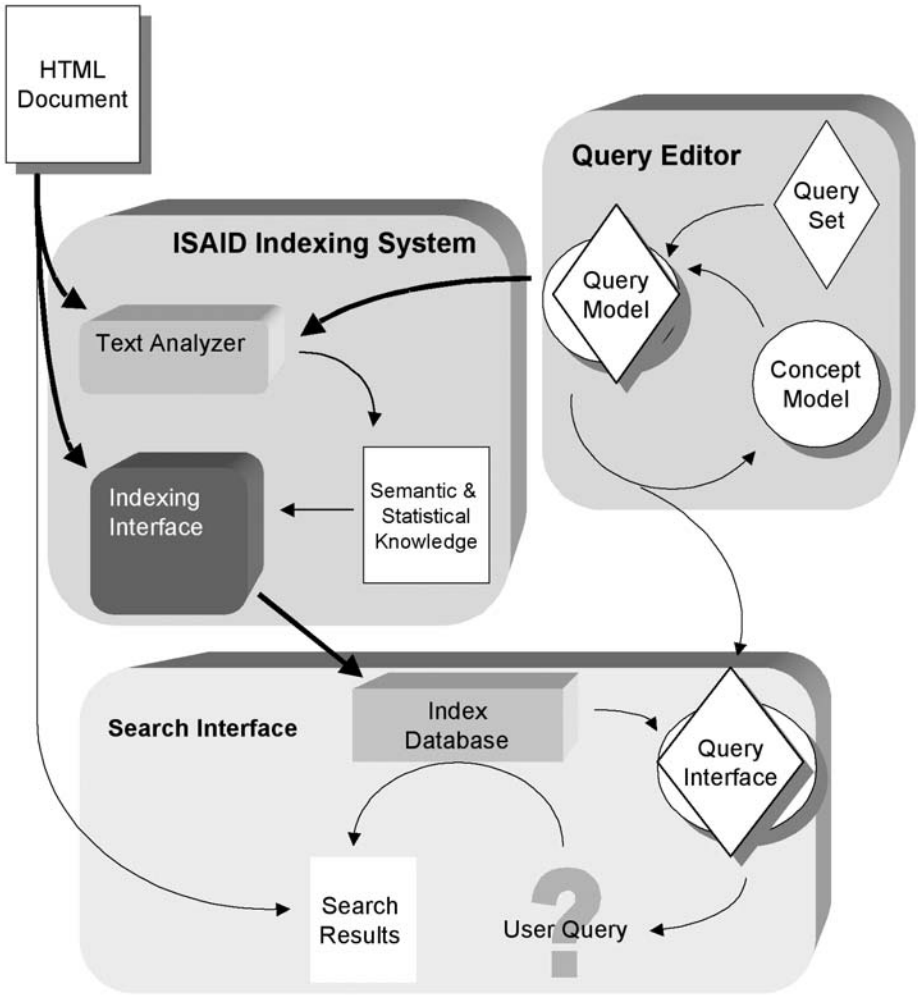


Figure 1 The ISAID indexing system's role in ELBook, a high precision information retrieval system. ISAID requires syntactic, semantic, and other domain-specific knowledge from query and concept models. Typically, a domain expert will first author a query model using the QueryEditor (*upper right*). The domain expert then exports the query model in XML and HTML formats for use in ISAID (*upper left*) and a search interface (*bottom*), respectively. Next, a system administrator will prepare one or more documents for indexing using the ISAID Text Analyzer. Then, one or more users of the ISAID indexing interface (Figure 2) can generate indexes for these documents, which are stored in a networked Index Database. Once such indexes have been generated, an end-user can search them using a dynamically generated Query Interface. Arrows indicate data flow: heavy arrows represent data input to and output from ISAID.

to navigate these documents and generate and store indexes for them. A query model guides the analysis of documents and provides templates for indexes selected by users of the indexing interface. A modified vector-space model of the statistical and semantic knowledge stored by the TA is used to propose indexes for individual document elements.

ELBook Query Model

In an ideal situation for precisely retrieving information, we would enumerate all the possible queries that users of a collection might ask, and then would index the entire collection with respect to those queries. Unfortunately, many collections have a huge diversity of readers; anticipating every possible information need of these readers is difficult. We generated a set of clinical generic queries, a "top-level query model" (TLQM), for use as a boiler-plate for the creation of diverse and more specific query models.[17] These generic queries resemble those that Cimino[18] developed to aid users in generating MEDLINE queries, those Pratt[19] designed for use in her system for categorizing MEDLINE search results, DynaCat, and those that Ely et al. modeled based on solicitations from practicing physicians.[20,21] Each generic query consists of a set of concepts combined through fixed text segments. Each query concept restricts query terms to those from a list of concept values.

The development of the TLQM has already been detailed extensively.[17] Briefly, we collected queries for information posed by clinicians in the practice of inpatient and outpatient internal medicine. We re-created the "*is-a*" portion of the UMLS semantic network (i.e., that portion of the semantic network that relates concepts by the *is-a* binary relation) in the concept hierarchy of a Protégé-2000 ontology. We then proceeded to identify these classes of concepts in the queries that we had collected and combined queries when possible over these classes. We augmented the hierarchy with additional abstract superclasses that we deemed nec-

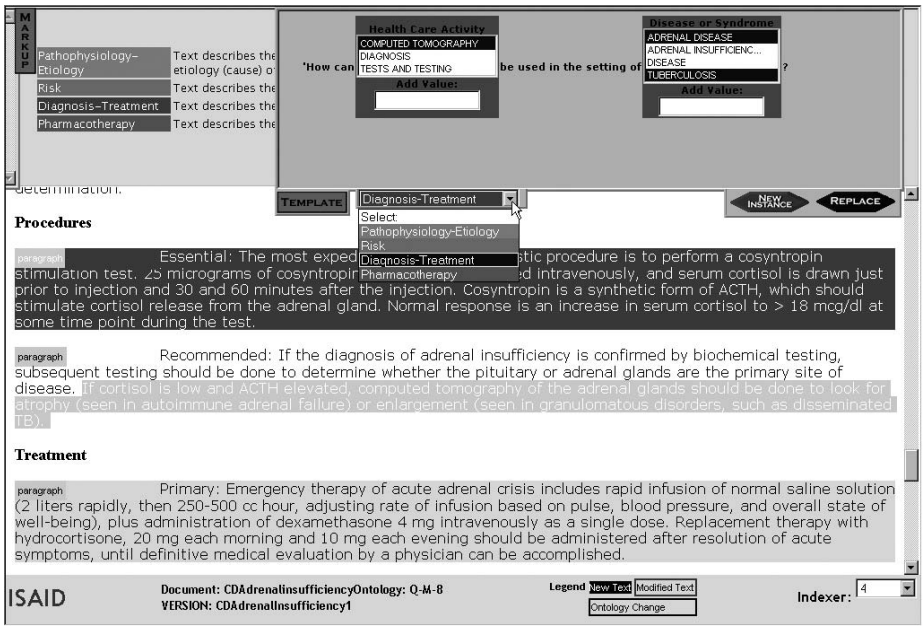


Figure 2 The ISAID indexing interface. The interface consists of a main window with four frames. At the top of the window is the query model frame and at the bottom is the document view frame. The two remaining frames, the markup view frame and the query template frame, have tabs (labeled "markup" and "template") and can be "pushed off" the display. The query template frame (magnified in B) displays which generic query and which concept values ISAID suggests for indexing the document element shown in B.

essary in order to minimize the number of generic queries in the TLQM. For example, we created the superclass "Manifestation" to encompass the semantic types *sign or symptom* and *anatomic abnormality*, both manifestations of disease processes. For each superclass, we restricted the corresponding set of allowed concept values in generic queries that contain the superclass to those UMLS Metathesaurus concepts whose semantic type is a member of the superclass.

Using the TLQM as a starting point, we built a textbook query model (TQM), a model of what we deemed would be the most common information needs of physicians turning to medical textbooks. This model contained only four generic queries (Table 2). We created the TQM by first designing a plug-in software component for Protégé-2000, a knowledge acquisition tool (22), called the Query Editor (see Figure 1). This component allows domain experts to create generic queries for use in ELBook. After creating these queries, Query Editor users can then export them in HTML or XML, along with semantic and statistical knowledge about the concepts in these prototypes (e.g., where the concepts occur in a concept hierarchy, the frequency of a concept in a set of query templates). ISAID later uses this knowledge to help propose indexes for documents (see below). We first re-created the TLQM in the Query Editor, and then proceeded to modify each generic query to create the TQM. In designing the TQM, we had two goals: to maximize the difference between queries so that the query vectors would appear as distinct as possible in the ISAID vector-space model (see below) and to preserve the ability of the TQM to satisfy the information needs of medical textbook readers. Thus, for the TQM, we combined the queries of the TLQM as necessary to generate the fewest number of maximally distinct queries that would still capture a large number of the queries in the TLQM query space. For example, we combined the TLQM queries "How can <DIAGNOSTIC PROCEDURE> be used to diagnose <DISEASE OR SYNDROME>"? and "How can <THERAPEUTIC OR PREVENTIVE PROCEDURE> be used to treat <DISEASE OR SYNDROME>"? into the single TQM query "How can <HEALTH CARE ACTIVITY> be used for <DISEASE OR SYNDROME>"? Since the same questions in the space of these two TLQM query templates occur in the space represented by the single TQM query template, this operation at once reduced the number of query templates from which ISAID must choose to suggest indexes, increased the difference between query templates in the query model, and preserved the ability of ISAID users to index the knowledge represented by the two original query templates.

ISAID Indexing Interface

The ISAID indexing interface consists of PERL CGI (Common Gateway Interface) programs that allow indexers to navigate and view documents, and to view, create, delete, or replace indexes for documents (see Figure 2). Navigating and indexing documents

Table 2 ■

The Textbook Query Model (TQM) Consisting of Four Generic Queries

Label	Format	Target Text Description
Pathophysiology/Etiology	What is the {Pathophysiology/Etiology} of {Manifestation/Pathology}?	Text describes the set of physiological abnormalities that together comprise a disease or symptom OR text describes the etiology (cause) of a disease or disease manifestation.
Risk	What is the {Risk Measure} of {Disease/Syndrome} given {Manifestation/Investigation/Pathology}?	Text describes the risk (relative risk, odds, or other risk) of a disease given a named risk factor.
Diagnosis/Treatment	How can {Health Care Activity} be used in the setting of {Disease/Syndrome}?	Text describes the use of a diagnostic, laboratory, therapeutic, or other procedure in the setting of disease.
Pharmacotherapy	How can {Pharmacotherapy} be used to treat {Disease/Syndrome}?	Text describes the use of a pharmacologic agent to treat a disease.

using the ISAID indexing interface relies on the Document Object Model as implemented for HTML 4.0 documents. The interface uses the knowledge that the TA stores to propose generic queries and to propose concept values in those prototypes as indexes for elements of documents.

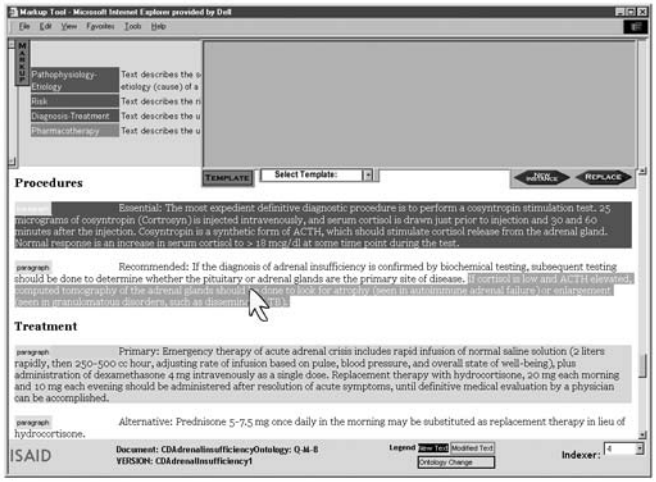
The interface consists of four frames: a query model frame, an indexing view frame, an index template frame, and a document navigation frame. The query model frame lists each of the query-prototypes in a chosen query model. The indexing view frame displays all the indexes associated with the current location in the document. The document navigation frame displays a scrollable view of the entire HTML document that a user is currently indexing. The ISAID TA provides the source HTML code for rendering the document, which the indexing interface modifies to show the query templates that users have indexed for all the elements in the document through color-coded background colors. The user can set the current indexing location to any element, or to an HTML container for an element (e.g., a paragraph that contains a sentence), by clicking on the element or a representation of its container.

The index template frame (Figure 3) requires the most complex user interaction of all the frames in the indexing interface. It has a "pull-down" list of all the generic queries in a given query model. Selecting a choice from this list displays the selected generic query as a template for an index, a combination of plain text and domain-specific semantic types and associated select lists of concept values. The indexing interface automatically selects and loads one of the generic queries as an indexing template. The frame also has two buttons that allow users to create a new instance of an index (i.e., index an element of the document with the completed template) or to replace one indexed template with another. Once a generic query is loaded as a template for an index, the user can select or de-select any values for any semantic type in the template at will (concept values proposed by ISAID are automatically selected when the template loads, see Figure 3). One or more values can also be added to the select list either by typing them into a text-input window underneath the concept labeled "Add Value" or by highlighting, dragging, and dropping text from the document into the same text-input window. Once satisfied with the choice of values for all the semantic types in the template, the indexer clicks the button labeled "New Instance." This starts a sequence of events that stores the data from the template in a relational database system (Oracle 8.5 Database, Oracle Corp, Inc.) and visually confirms the indexing process to the user.

As the data from the indexed generic query are written to the database, the indexing interface changes the background color of the text in the document to match the color of the generic query label shown in the query model frame (and also in the pull-down list of generic queries in the query template frame). If a user indexes a given location with more than one instance of a generic query, the indexing interface changes the background color of the location to gray. These changes provide visual cues to indexers as to exactly what portions of the document have already been indexed and with which generic queries. For each change in the templates with which a user indexes the current document location, the display of indexed templates and concept values is updated in the markup view frame.

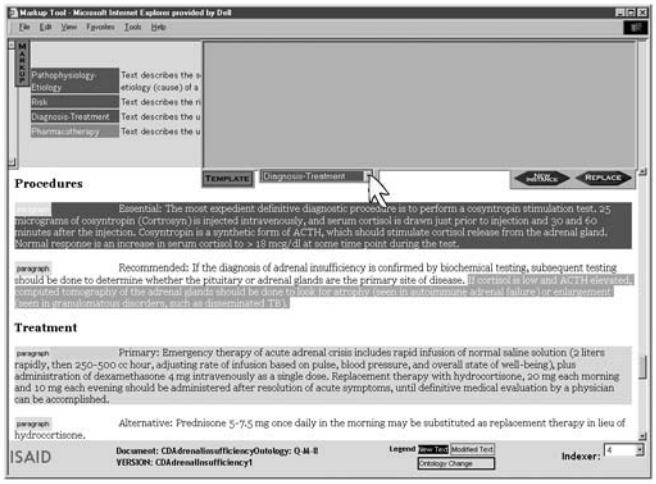
ISAID Text Analyzer

The TA creates and stores semantic and statistical knowledge about the document. The indexing inter-



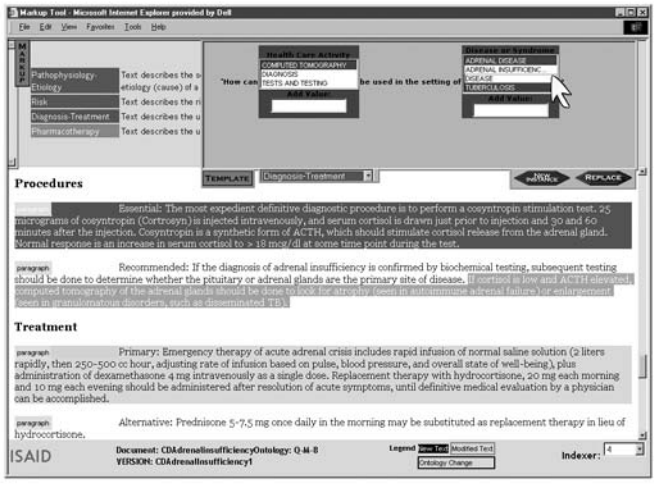
(A)

User selects text in document. The user can select a single sentence, paragraph, list item, whole list, table cell, table row, or whole table.



(B)

User chooses a query template with which to index the text from a pull-down list of the query templates in the query model.



(C)

User makes selections in query template. The ISAID system will propose (by highlighting) certain selections in the template based on the content of the text. The user can de-select any of these and make new selections at will see (Fig. 3D)).

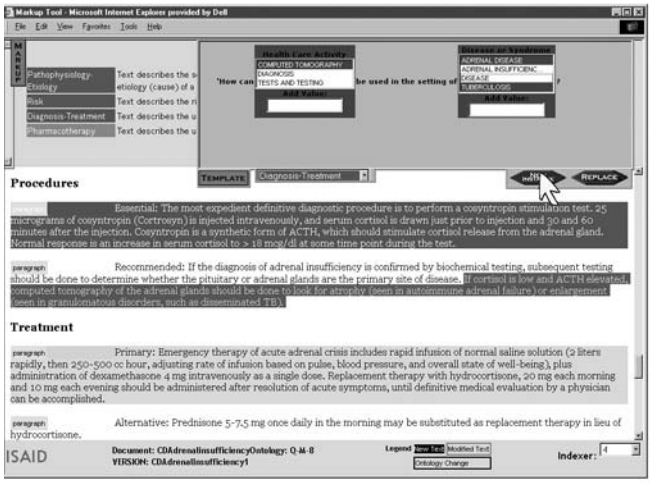
Figure 3 *A–F*, Interaction with the ISAID indexing interface. The steps shown are those a user would take to create a new instance of an index for a sentence from the Special Operations Forces Handbook of Medicine (U.S. Armed Forces, used with permission).

face taps this knowledge later to present choices of indexes to its users. The TA identifies and labels HTML document elements, and performs syntactic, semantic, and statistical analysis of the text. HTML element labeling is essential so that users can navigate documents during the indexing process. The syntactic, semantic, and statistical analyses form the basis for proposing accurate indexes for documents.

(D)

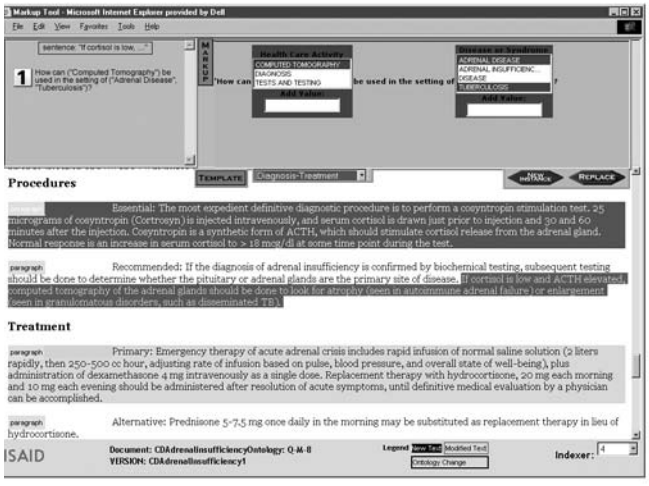
User interaction with the query template frame. The frame displays a single query template with a list of possible text values for each concept. Some of these values are automatically selected by ISAID. The user can change any of these selections at will. The user can also type in new values (or drag text values into the list from the text of the document).

Figure 3 (*continued*)



(E)

User commits index. By clicking on the "New Instance" button, the user commits his selection of query template and template concept values, which are then stored in the indexing database, associated with the text of the document.



(F)

ISAID interface reflects indexing actions. To clue the user that the index has been stored, the text background color changes to match that of the chosen query template. Also, an entry is made in the markup view frame.

The TA begins by converting HTML document elements to plain (ASCII) text. It then identifies multiword UMLS concepts and syntactically tags the terms of the text using commercially available software tools.[23,24] The TA then dynamically obtains and locally stores the semantic types of all nouns and noun-modifiers it has identified in the text. It ascertains these semantic types using a first match algorithm and the UMLS web API.[25]

The TA also uses a simple HTML heading context model to store context knowledge in the document, which is later used by the indexing interface as one source of concept values it proposes to indexers. Pur-

cell developed a fixed context model for clinical trial journal articles that was applied by hand to documents. Our context model is more general, based on HTML document structure (specifically, the content of title and section heading elements), and inferred automatically by the TA. The value of this model depends on the extent to which authors use terms in HTML headings that have meaning in the domain of interest. For example, while the headings H1: "Section 1," H2: "Section 1A," and so on may provide structure for a document, they are unlikely to help either manual or automated indexers to determine the appropriate concepts with which to index elements of the document. On the other hand, sections labeled with more meaningful terms (i.e., pathophysiology, diagnosis, and therapy) are of greater usefulness to indexers and can be a valuable source of knowledge for indexing elements within a document.

Finally, the TA generates summary statistics based on its semantic analysis of documents. ISAID uses these statistics in a modified vector-space model (see below) to propose generic queries as indexes. For each semantic type in a given query model, the TA tabulates and stores the number of document elements (sentences, list items, and table cells) in which it occurs.

Vector-Space Model Adapted for Automated Indexing

We adapted the term-vector space model,[26] commonly used to find documents whose term vectors are closest to a given query vector, to find instead those queries a given document is most likely to answer (Figure 4). This novel application of the model also used somewhat different definitions of "documents," "queries," and "terms" and required new methods for the appropriate calculation of "term" weights.

In many IR systems, documents are defined by physical and/or spatial separation in source documents (e.g., individual journal articles, separate web URLs). For example, although a user may only be interested in one section of an NLM-indexed document, the entire document is indexed as a unit and, if appropriate, returned in search results. The ISAID system strives to index much more specific passages of text and has no a priori definition of individual documents. However, since a single sentence may answer any given query we defined a document to be any sentence, list item, or table cell. Using this new definition, we calculated document frequency for a given term based on how many sentences, list items, or table cells of the text contain the term. Furthermore, the "terms" in ISAID query templates were constrained to UMLS Semantic Network semantic types. We therefore constrained "terms" in our vector-space model to these same semantic types, and, for the vector space model only, regarded a "query" as the set of these semantic types that compose each query template in the query model.

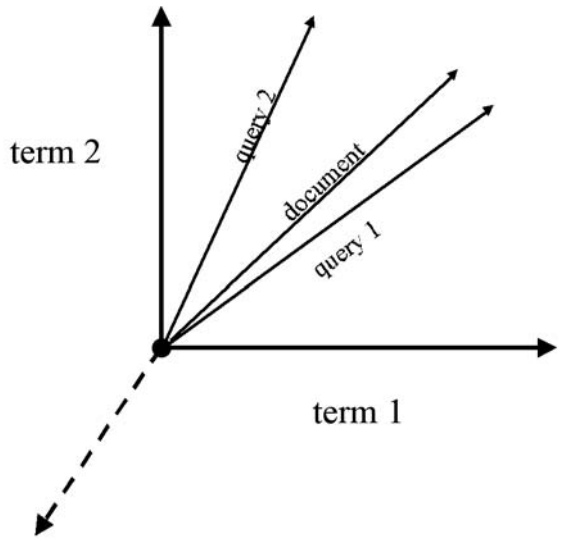


Figure 4 The term-vector-space model adapted for information indexing. A given document is compared with several queries (only two queries are shown) using each component of a term vector (only two components are shown).

The simplest term-vector-space models assess how similar a given document, d, and query, q, are with respect to a given term, ti, based on term frequency (tf_i), a measure of the frequency with which ti occurs in d, and inverse document frequency (idf_i), a measure of how frequently ti occurs in the entire document collection. However, documents as small as a single sentence are unlikely to contain multiple occurrences of t_i, thus tf_i is not likely to vary much over terms and documents or a set of generic queries. We therefore chose base model vector weights on binary indicators of term frequency.

Furthermore, there is significant statistical information contained in each ISAID query model. We defined a statistic analogous to idf_i to help capture how much a given generic query is "about" a term t_i, the inverse query frequency, iqf_i:

$$iqf_i = \log\left(\frac{\text{no. of queries}}{\text{no. of queries that contain term } t_i}\right) + 1$$

The formula for calculating the statistic is identical to that for calculating idf_i, except that it is based on the fraction of queries that contain term t_i, instead of the fraction of documents that contain term t_i. The inverse query frequency statistic is calculated for each semantic type when a user exports a query model using the Query Editor.

To yield the components of each vector in the term-vector-space model we adapted for information indexing, we multiplied the *idf* and *iqf* statistics. Intuitively, concepts that appear rarely in a document collection and/or in the query model are given a higher weight component for that term to reflect a greater imperative for indexing them in documents, and a greater ability of the concepts to discriminate between possible indexed queries. We then compute vectors for each generic query in the query model and for a given set of documents. The components of each vector are the "term" weights for each concept that occurs in the query model, defined as *iqf*idf* using binary term frequency indicators. We can then compute the query vector that best matches a document vector using a cosine closeness measure:

$$\frac{\sum_{k=1}^{t} (type_{ik} \bullet qtype_{jk})}{\sqrt{\sum_{k=1}^{t} (\text{type}_{ik})^2 + \sum_{k=1}^{t} (qtype_{jk})^2}}$$

where t is the union of types in the document and query, $type_{ik}$ is the magnitude of dimension k in the document vector i, and $qtype_{jk}$ is the magnitude of dimension k in query vector j.

Evaluation

Experimental Controls: Pilot Evaluation

The issue of appropriate controls in the evaluation of a system like ISAID is a difficult one. Other than pencil and paper or simple word processors, there are no current methods for generating the type of document indexes that ISAID can. We suspected the use of such "manual" indexing methods would be quite time-consuming and tedious for subjects. To confirm our suspicions, we performed two pilot evaluations of ISAID compared with manual indexing methods. The editor of an infectious disease textbook developed a specialized query model for use in the two studies, and was himself one of the subjects in each study. In the first study, the editor generated indexes using pencil and paper by recording instances of query templates (identified by number) and concept values to complete the template in the margins of a chapter from the textbook on the treatment of infections due to *Eikenella corrodens* (the chapter contained 2,384 words, 113 sentences, and 19 paragraphs). For example, next to the margins of one paragraph, the author recorded "Query 4" (a drug susceptibility query template), and "*Eikenella corrodens*—nafcillin" (concepts for the completion of this template). A second subject, a family practice resident, then used an early version of the ISAID indexing interface to index the same chapter. In the second study, the editor and a resident physician in internal medicine used the same methods as in the first study to index a much longer chapter on the treatment of infections due to *Staphylococcus aureus* (6,151 words, 266 sentences, and 83 paragraphs).

The textbook editor using a manual indexing method on the *Eikenella corrodens* chapter required approximately three hours to generate by hand all the indexes that he deemed necessary to index the document. The resident required only 42 minutes using the ISAID indexing interface. In the second comparison (using a much longer textbook chapter), the resident, who used the ISAID indexing interface, required 186 minutes to index the entire chapter, and generated 264 instances of indexed generic queries. The editor estimated his total time to index the same chapter to be over 8 hours (he reported he found the task so tedious to perform by hand that could not index the chapter in one session). The editor generated 186 instances of generic queries as indexes, and stopped indexing approximately three quarters through the chapter due to fatigue.

Because the results of the pilot evaluations suggested ISAID was clearly faster than such methods, we chose to devote our resources to a more detailed evaluation of the system's index proposal methods that compared different features of ISAID.

Experimental Design: Methods Evaluation

We attempted to ascertain the value to users of the natural-language-processing methods that the ISAID indexing interface uses to propose concept values and select generic queries as templates for indexes. We suspected individuals using ISAID would vary in the time they require to index any given material. We therefore opted for a repeated measure (time series) experimental design, in which each of twelve subjects used three nearly identical versions of ISAID. This design controls for large variations in indexing times

between subjects (between subject effects). Each subject used the ISAID indexing interface with both automated concept highlighting and generic query selection (CP), the interface with only concept highlighting (C), and the interface without automated concept highlighting or generic query selection (NCNP). The CP version automatically loaded a generic query into the query template frame (see Figure 3b) when a new location in the document was selected. In the two other versions of the interface, the user was required to make a selection of generic query from the pull-down menu in the frame. Furthermore, in the NCNP version, the same concepts were shown as in the two other versions, but without any automatic selection (indicated by highlighted values); the user was required to select the concepts he or she deemed appropriate for indexing. Even though we had planned to use a short training/run-in phase to reduce learning effects, we recognized subjects would likely still index documents faster with more and more experience using the interface. Therefore we randomly assigned subjects to use the three versions of the interface each in a different order. We replicated the design of the experiment once for a total of twelve subjects.

Each of the twelve subjects indexed the same three documents in the same order. To create the documents used in these indexing experiments, we selected sections from three different medical textbooks in electronic form and deleted paragraphs, lists, and/or tables to make them of approximately equal length (about 2,000 words each). We made the selections from each textbook completely arbitrarily (we did not use random selection methods, but we did not examine the performance of the TA on these documents prior to selecting them). We created the first document from a chapter on adrenal insufficiency in a military handbook of medicine, the second from a section concerning liver diseases in *The Washington Manual of Medical Therapeutics*,[27] and the third from a chapter on legionellosis in *Cecil Textbook of Medicine*.[28]

Subjects

We recruited twelve physicians (all housestaff) from a tertiary care medical center as subjects using online advertisements and e-mail lists. Subjects were required to have at least some residency training in clinical medicine or surgery and some experience using a web browser. We paid each subject a flat fee for participating in the study (in particular, this sum was not based on length of participation).

Training/Run-in Phase

We trained subjects to use the indexing interface, which was essentially the same for each of three versions except for highlighting of concept values and selection of a query template by the interface. Each subject spent 15 minutes reviewing a self-guided, animated tutorial that explained the purpose of the indexing interface and exactly how each of the frames, buttons, and select lists functioned. The subjects could view or review any portion of the tutorial at will. After the self-guided tutorial, subjects spent another 15 minutes reviewing guidelines for using each of the four query templates and using the fully enabled version of the interface to complete five indexing tasks using a test document. Finally, we solicited and answered any questions that subjects had regarding the use of the interface.

Data Collection

While subjects were using each version of the interface, the interface logged some of their actions automatically. For five actions (change current location, load query template, create, replace, or delete an instance of an indexed query template) the interface noted and recorded the time of the action. We truncated subjects' indexing sessions to approximately 45 minutes (for a maximum of two and quarter hours of indexing per subject, although several subjects finished the assigned indexing tasks in much less time). Subjects were not required to index the elements of documents in any particular order. All but one of the subjects completed training and the three assigned indexing tasks on one day. One subject indexed two and almost all of the third document on one day, but, because the local network file system became unavailable, had to return a week later to finish the third indexing session. We combined the times recorded in the two log files for the third indexing session of this individual.

Outcomes of Interest

We were interested in the ability of each version of the indexing interface to increase the speed with which users create accurate indexes. It is imperative to consider both speed and accuracy as outcome measures, because neither the fast creation of inaccurate indexes nor the slow creation of accurate indexes is desirable. Because subjects were allowed to index a document at will during each indexing session, it was not possible to determine how much of each document they had indexed. Instead, we measured speed by calculating the number of index tuples each sub-

ject generated using the indexing interface and dividing by the time required to generate the tuples. We defined an index tuple as a set of concept values contained in a single query template that a subject generates as an index for a specific location in a document. If a subject selects more than one concept value, each value is placed into a different tuple. For example, the *diagnosis-treatment* query template contains two concepts: *health-care activity* and *disease or syndrome*. Suppose a user indexed paragraph 2, sentence 1 of a document with a diagnosis-treatment query template and selected the values for health-care activity, magnetic resonance imaging (*MRI*) and ultrasound (*US*), and two values for disease or syndrome, *liver disease* and *hepatic disease*. Our method would explode this single query template into four index tuples (MRI-liver disease, MRI-hepatic disease, US-liver disease, and US-hepatic disease). We then calculated the number of these tuples generated as indexes by each subject per minute (TPM).

Because we lacked a gold standard with which to compare the indexes that subjects generated, we measured accuracy by using consensus among the indexers. We used concordance (the amount of agreement between a given subject and the rest of the subjects), examining all index tuples generated, tuples generated by at least two subjects, and tuples generated by at least three subjects.

We combined concordance and TPM into one measure: CTPM, the number of index tuples created per minute with a given concordance between subjects. Because subjects could index an HTML element or its container (or container's container), we expanded the search for concordant index tuples for HTML elements to include tuples indexed for their containers. For example, a tuple created by subject A for paragraph 2, sentence 1 would be in concordance with an identical tuple created by subject B as an index for all of paragraph 2.

We also examined how consistent the twelve indexers were with each other using a form of Hooper's measure.[29] For each instance of an index tuple generated by either member of a given pair of indexers, we tabulated the number of concordant (t_c) and discordant (t_d) terms, and then summed these tabulations for each pair over each of the three documents they indexed. We calculated consistency as:

$$Consistency(A,B) = \frac{t_c}{t_c + t_d} \bullet 100$$

Partly because subjects were not required to index document elements in any particular order, there were numerous elements for which only one indexer of a pair had assigned any indexed terms. We did not include such terms in our consistency calculations. However, these tuples were included in the indexing rate and concordance analysis.

Usability Survey

After each of the twelve subjects used ISAID to index the three textbook chapters, we asked them to complete a very short survey concerning their experience with the tool. We informed them that their responses would be anonymous and confidential and would not impact payment for their time. The survey consisted of four questions requiring responses on a scale of 1 to 5, and an estimation of the percentage of important clinical knowledge the subjects felt they could index using ISAID.

Results

Indexer Speed and Consistency

The twelve subjects generated on average 140 tuples of indexing in each of the 45-minute indexing sessions (minimum: 10; maximum: 861). The average number of tuples subjects generated per minute (TPM) was 4.4 (minimum: 0.5; maximum: 23). The rate subjects generated tuples increased from 3.2 for the first indexing session, to 3.6 for the second indexing session, and to 6.3 for the final indexing session, but this increase was not statistically significant ($p > 0.05$ by repeated measures ANOVA). There were 66 possible pair-wise comparisons of consistency between the twelve subjects. Consistency by Hooper's measure ranged from 15% to 65% with a mean of 41%, 31%, and 40% for each of the documents. Consistency did not vary widely by version of ISAID (CP: 41%; NC: 38%; NCNP: 35%).

Index Proposal Methods

TPM was highest when subjects used the full version of the indexing interface (CP, 5.6; std. dev.: 5.6) compared with when they used the interface that only proposed concept values (C, 3.9; std. dev.: 1.9) and with when they used the interface that did not propose concept values or select query templates (NCNP, 3.9; std. dev.: 4.0). However this result was also not statistically significant using a repeated measures ANOVA analysis ($p = 0.49$).

Indexing rate fell dramatically for all three versions of the indexing interface with increasing concordance

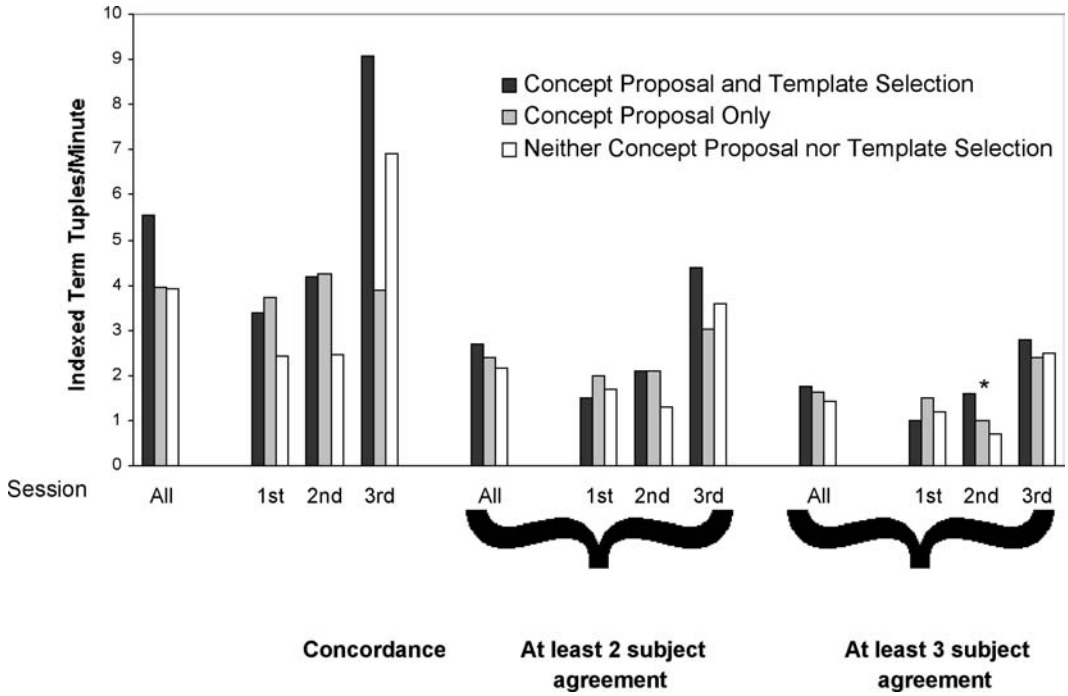


Figure 5 The average rate (in indexed term tuples per minute) that subjects generated index tuples using three versions of the indexing interface: one that proposes concept values, one that proposes concept values and selects query templates, and one that neither proposes concept values nor selects query templates, by concordance index and by indexing session. See Methods for calculation of concordance index. *Statistically significant in ANOVA analysis, $p < 0.05$.

(Figure 5). Using data from all the indexing sessions, subjects using the CP version of the indexing interface generated more index tuples per minute compared to when they used either of the other two versions for all levels of concordance. However, none of these comparisons was statistically significant by repeated measures ANOVA (for $p = 0.05$).

We considered the possibility that, despite attempts to train them during a run-in phase, the subjects may not have been fully trained until sometime during the actual experiment. Indeed, subjects using the full version of ISAID improved the speed and accuracy with which they indexed documents from the first to the second and from the second to the third index sessions by all of the outcomes measures. We analyzed the effect of assigned version of the indexing interface on these measures during each of the three indexing sessions. The power to detect differences between the interfaces is low in this simple ANOVA analysis because the number of subjects using each version drops from twelve to four.

There were no statistically significant differences between subjects by version of the indexing interface during the first or third indexing session using unadjusted or concordance-adjusted outcome measures. During the second indexing session, users assigned to the CP version of the indexing interface were statistically significantly faster and more accurate than those assigned to the other two versions for three-subject concordance ($p < 0.05$). Comparisons using all the other outcome measures were not statistically significant.

Usability Survey

We received responses to all the questions on a brief usability survey from all twelve of the subjects (Table 3). Subjects answered the first four questions regarding their understanding and use of ISAID by responding on a five point scale, in which "1" was the most unfavorable and "5" the most favorable response. We asked subjects to rate how well they understood what the tool was designed to do. All subjects rated their understanding a "3," "4," or "5," with a mean of 4.1. We also asked the subjects to rate their comfort using the tool. Again, no subject responded to the question with a rating less than "3," and the mean was 4.0. Next, we asked subjects to rate how difficult the indexing interface was to use, with specific reference to the layout of items in the inter-

Table 3 ■

Results of a Brief Usability Survey of the ISAID Indexing System.

Item	Scale	Range	Mean
1. Do you think you understood what the ISAID tool is designed for?	1 (no)–5 (yes)	3–5	4.2
2. How comfortable do you feel now using the ISAID Indexing Tool?	1 (very uncomfortable)–5 (very comfortable)	3–5	4.0
3. Do you think the ISAID interface (how items were laid out) is:	1 (very difficult to use)–5 (very easy to use)	3–4	3.5
4. How easy would it be for you to use the ISAID indexing tool now on other textbook chapters?	1 (very difficult)–5 (very easy)	3–5	4.2
5. What is your estimate of the amount of important clinical knowledge in text you were able to index using ISAID?	0–100%	60–90%	78%

face. All subject responded either "3" or "4," with a mean rating of 3.5. We also asked the subjects how easy it would be to use the indexing interface on other textbook chapters. Again subjects responded only "3," "4," or "5," with a mean rating of 4.2. Finally, we asked subjects what their estimate of the amount of important clinical knowledge they were able to index using ISAID, from 0% to 100%. This last measure should reflect the subjects' assessment of the appropriateness and completeness of the TQM, as well as their ability to use ISAID . The mean response was 78%, ranging from 60% to 90%.

Discussion

The results of our studies comparing ISAID to manual indexing methods indicate that users of the computer-based ISAID indexing interface could index textbook chapters several times faster than indexers who use manual methods. The indexing interface performs numerous functions that speed the process of indexing, including proposing concept values, selecting query templates, providing a scrolling, windowed view on documents and providing visual cues in the documents to users as they generate indexes.

The consistency rates we observed between ISAID indexers (30–40%) approximate those observed between NLM indexers.[30] However there are significant differences between the two indexing tasks. NLM indexers choose relatively few terms from a controlled vocabulary to index an entire journal article; whereas ISAID indexers can select any term as indexes, and often have to make these selections dozens of times for a single document. Despite this greater indexing freedom, ISAID indexers agreed on indexing terms quite often. This could have been due to an implicit restriction of terms imposed by the indexing interface; subjects may have been reluctant to choose index terms that were not automatically suggested. However, 47% of the unique concept values that subjects chose as indexes were not suggested by ISAID, either because they did not occur explicitly in the documents (23%), or were not detected by the TA (24%). Thus, ISAID indexers were surprisingly concordant given the amount of freedom they had to choose indexes.

On average, subjects created index tuples faster using the full version of the indexing interface (that proposed concept values and generic queries) ignoring concordance and for two- and three-subject concordance. In addition, subjects' combined indexing speed and accuracy improved over the course of the experiment, indicating substantial learning effects, despite the use of a training/run-in phase. Results from the second indexing session did reach statistical significance for three-subject concordance. Furthermore, results from the third indexing sessions consistently showed that users of the full version of the indexing interface generated more, concordant index tuples per minute by all outcome measures. Collectively, these results suggest a larger study of ISAID's index proposal methods would show a statistically significant benefit to indexers in terms of indexing speed and concordance.

Of interest, subjects using the version of ISAID that only proposed concepts actually performed more poorly than subjects using the version that did not propose concepts during the third indexing session. One possible explanation is that the suggestion of concepts for indexing without appropriately relating those concepts (by proposing the correct query template) might confuse indexers.

The indexing interface is quite complex and presents users with a large amount of information. It is reasonable to expect that users find the indexing inter-

face more challenging to use than other web-based tools they are perhaps familiar with, such as PubMed's interface. Still, the results of the usability survey suggest that subjects understood the intended purpose of the indexing interface and felt fairly comfortable using the tool. They felt they could use the indexing interface to capture almost 75% of the clinical knowledge contained in the chapters that they felt was important. They also felt they could continue to use the tool fairly easily to index more textbook chapters. The twelve subjects felt that the interface could have been easier to use, which suggests that developing improvements to the indexing interface should remain a high priority.

ISAID requires complex input in the form of a query model and structured HTML documents. There could conceivably be important interaction effects in our experimental results due to the choice of these inputs. It would be important to examine any such effects by replicating our experiment using a substantially different query model and/or set of documents.

Highly precise searches using current information retrieval systems, from PubMed to Altavista, are rare. Attempts to improve the precision of searches that use these systems, by and large, have failed. The ELBook system could provide highly precise searches of documents, if the type of fine-grained indexing that the system requires could be generated expeditiously. The ISAID indexing system is a significant step towards the automated creation of complex, precise indexing for full-text documents with a minimum of human guidance.

The authors thank Apelon Technologies; Victor L. Yu, MD, of the University of Pittsburgh, Department of Infectious Disease; Todd Ferris, MD, Stanford Medical Informatics; Andrew Kehler, PhD, Department of Linguistics, University of California, San Diego, CA; and Ms. Carol Maxwell, Stanford University.

References ■

1. Forsythe DE, Buchanan BG, Osheroff JA, Miller RA. Expanding the concept of medical information: An observational study of physicians' information needs. Comput Biomed Res. 1992;25(2):181–200.
2. Covell DG, Uman GC, Manning PR. Information needs in office practice: are they being met? Ann Intern Med. 1985; 103:596–9.
3. Giuse NB, Huber JT, Giuse DA, et al. Information needs of health care professionals in an AIDS outpatient clinic as determined by chart review [see comments]. J Am Med Inform Assoc. 1994;1(5):395–403.
4. Chambliss ML, Conley J. Answering clinical questions. J Fam Pract. 1996;43(2):140–4.
5. Hersh WR, Hickam DH. An evaluation of interactive Boolean and natural language searching with an online medical textbook. J Am Soc Info Sci. 1995;46(7):478–9.
6. Nicholls J, Howes M, Jones R. Information-seeking behaviour using paper and electronic versions of a textbook. In IEE Colloquium on Human-Computer Interface Design for Multimedia Electronic Books (Digest No.1995/038). London, IEE, 1995, pp 5/1–3.
7. Fagan LM, Berrios DC, Chan A, et al. Information Retrieval Using UMLS-based Structured Queries. Proc AMIA Annu Fall Symp. 2001, p 902.
8. Bernstein LM, Siegel ER, Goldstein CM. The hepatitis knowledge base. A prototype information transfer system. Ann Intern Med. 1980;93(1 Pt 2):169–81.
9. Purcell GP, Shortliffe EH. Contextual models of clinical publications for enhancing retrieval from full-text databases. In: Gardner RM (ed), Nineteenth Annual Symposium on Computer Applications in Medical Care. Philadelphia, Hanley & Belfus, 1995, p 851–7.
10. Berrios DC. Automated indexing for full text information retrieval. Proc AMIA Symp. 2000:71–5.
11. Humphrey SM. Interactive knowledge-based indexing: the MedIndEx system. In RIAO 88 Program. Conference with Presentation of Prototypes and Operational Demonstrations: User-Oriented Content-Based Text and Image Handling. Paris, CID, 1988, pp 883–98, vol. 2.
12. Aronson AR, Bodenreider O, Chang HF, et al. The NLM Indexing Initiative. Proc AMIA Symp. 2000:17–21.
13. Jain NL, Friedman C. Identification of findings suspicious for breast cancer based on natural language processing of mammogram reports. Proc AMIA Annu Fall Symp. 1997:829–33.
14. Friedman C. A broad-coverage natural language processing system. Proc AMIA Symp. 2000:270–4.
15. Hripcsak G, Kuperman GJ, Friedman C. Extracting findings from narrative reports: software transferability and sources of physician disagreement. Methods Inf Med. 1998; 37(1):1–7.
16. Jain NL, Knirsch CA, Friedman C, Hripcsak G. Identification of suspected tuberculosis patients based on natural language processing of chest radiograph reports. Proc AMIA Annu Fall Symp. 1996:542–6.
17. Cucina RJ, Shah MK, Berrios DC, Fagan LM. Empirical Formulation of a Generic Query Set for Clinical Information Retrieval Systems. In MedInfo 2001.
18. Cimino JJ, Aguirre A, Johson SB, Peng P. Generic queries for meeting clinical information needs. Bulletin of the Medical Library Association. 1993;81(2):195–206.
19. Pratt W. Dynamic organization of search results using the UMLS. Proc AMIA Symp. 1997:480–4.
20. Ely JW, Osheroff JA, Gorman PN, et al. A taxonomy of generic clinical questions: classification study. BMJ. 2000; 321(7258):429–32.
21. Ely JW, Osheroff JA, Ebell MH, et al. Analysis of questions asked by family doctors regarding patient care. BMJ. 1999; 319(7206):358–61.
22. Noy NF, Sintek M, Decker S, et al. Creating Semantic Web contents with Protege-2000. IEEE Intelligent Systems. 2001; 16(2):60–71.
23. Tuttle MS, Olson NE, Keck KD, et al. Metaphrase: an aid to the clinical conceptualization and formalization of patient problems in healthcare enterprises. Methods Inf Med. 1998; 37(4–5):373–83.
24. Soderland S, Aronow D, Fisher D, et al. Machine Learning of Text Analysis Rules for Clinical Records. In TE-39: University

of Massachusetts, Center for Intelligent Information Retrieval Technical Report. Boston, University of Massachusetts, 1995.
25. Berrios DC, Kehler A, Fagan LM. Knowledge Requirements for Automated Inference of Medical Textbook Markup. Proc Amia Symp. 1999:676–80.
26. Salton G, Buckley C. Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management. 1988;24(5):513–23.
27. Ahya SN, Kellie, K. F, Paranjothi S, et al. (eds). The Washington Manual of Medical Therapeutics. Philadelphia, Lippincott Williams & Wilkins Publishers, 2001.
28. Cecil RL, Bennett JC, Goldman L (eds). Cecil Textbook of Medicine. Philadelphia, W.B. Saunders, 2000.
29. Hooper RS. Indexer consistency tests: origin, measurement, results, and utilization. Bethesda, MD, IBM Corporation, 1965 [Report No.: TR95-56].
30. Funk ME, Reid CA. Indexing consistency in MEDLINE. Bulletin of the Medical Library Association. 1983;71(2):176–83.
31. Salton G. A comparison between manual and automatic indexing methods. Am Doc. 1969;20(1):61–71.